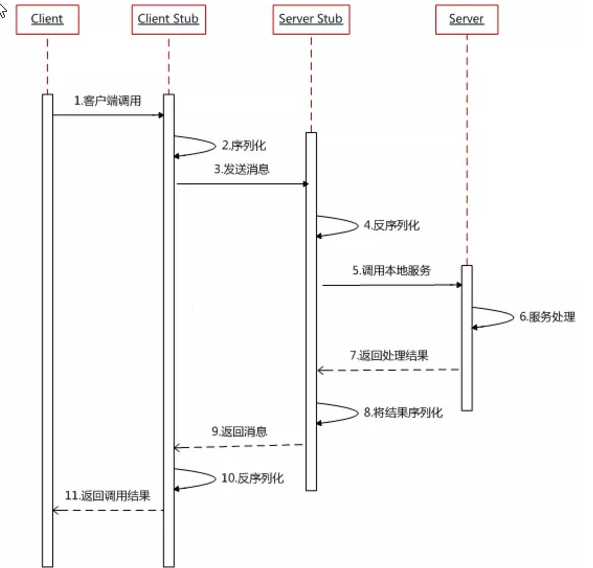
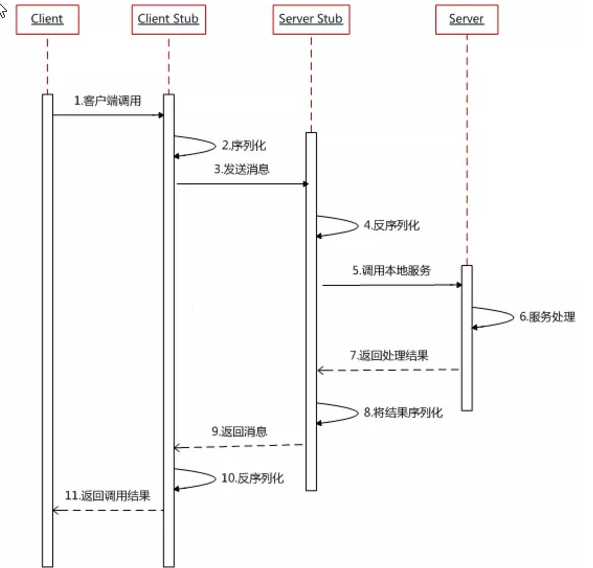
# RPC：



RPC（Remote Procedure Call）—远程过程调用，它是一种通过网络从远程计算机程序上请求服务，而不需要了解底层网络技术的协议。



## 既有 HTTP ,为啥用 RPC 进行服务调用?

###RPC 只是一种设计而已

RPC 只是一种概念、一种设计，就是为了解决 **不同服务之间的调用问题**, 它一般会包含有 **传输协议** 和 **序列化协议** 这两个。

但是，HTTP 是一种协议，RPC框架可以使用 HTTP协议作为传输协议或者直接使用TCP作为传输协议，使用不同的协议一般也是为了适应不同的场景。

# 分布式系统的经典基础理论

## 分布式系统设计两大思路：中心化和去中心化

### 中心化设计：

Master slave Hadoop

两个master

### 2) 去中心化设计：

节点自由选择中心，zookeeper，

注意脑裂

## 分布式与集群的区别是什么？

* **分布式：** 一个业务分拆多个子业务，部署在不同的服务器上
* **集群：** 同一个业务，部署在多个服务器上。比如之前做电商网站搭的redis集群以及solr集群都是属于将redis服务器提供的缓存服务以及solr服务器提供的搜索服务部署在多个服务器上以提高系统性能、并发量解决海量存储问题。

# CAP定理

一致性（Consistence） :所有节点访问同一份最新的数据副本： consistence 再ACID中对应于ACID， 一个用户请求，要么成功，要么失败。

可用性（Availability）:每次请求都能获取到非错的响应——但是不保证获取的数据为最新数据

分区容错性（Partition tolerance） : 分布式系统在遇到某节点或网络分区故障的时候，仍然能够对外提供满足一致性和可用性的服务。

**当发生网络分区的时候，如果我们要继续服务，那么强一致性和可用性只能2选1。也就是说当网络分区之后P是前提，决定了P之后才有C和A的选择。也就是说分区容错性（Partition tolerance）我们是必须要实现的。**

<http://www.cnblogs.com/hxsyl/p/4381980.html>

# BASE理论

BASE 是 Basically Available（基本可用） 、Soft-state（软状态） 和 Eventually Consistent（最终一致性） 三个短语的缩写。BASE理论是对CAP中一致性和可用性权衡的结果，其来源于对大规模互联网系统分布式实践的总结，是基于CAP定理逐步演化而来的，它大大降低了我们对系统的要求。

## BASE理论的核心思想

即使无法做到强一致性，但每个应用都可以根据自身业务特点，采用适当的方式来使系统达到最终一致性。也就是牺牲数据的一致性来满足系统的高可用性，系统中一部分数据不可用或者不一致时，仍需要保持系统整体“主要可用”。

针对数据库领域，BASE思想的主要实现是对业务数据进行拆分，让不同的数据分布在不同的机器上，以提升系统的可用性，当前主要有以下两种做法：

按功能划分数据库

分片（如开源的Mycat、Amoeba等）。

由于拆分后会涉及分布式事务问题，所以eBay在该BASE论文中提到了如何用最终一致性的思路来实现高性能的分布式事务。

## BASE理论三要素

1. 基本可用

基本可用是指分布式系统在出现不可预知故障的时候，允许损失部分可用性。但是，这绝不等价于系统不可用。

比如：

响应时间上的损失:正常情况下，一个在线搜索引擎需要在0.5秒之内返回给用户相应的查询结果，但由于出现故障，查询结果的响应时间增加了1~2秒

系统功能上的损失：正常情况下，在一个电子商务网站上进行购物的时候，消费者几乎能够顺利完成每一笔订单，但是在一些节日大促购物高峰的时候，由于消费者的购物行为激增，为了保护购物系统的稳定性，部分消费者可能会被引导到一个降级页面

2. 软状态

软状态指允许系统中的数据存在中间状态，并认为该中间状态的存在不会影响系统的整体可用性，即允许系统在不同节点的数据副本之间进行数据同步的过程存在延时

3. 最终一致性

最终一致性强调的是系统中所有的数据副本，在经过一段时间的同步后，最终能够达到一个一致的状态。因此，最终一致性的本质是需要系统保证最终数据能够达到一致，而不需要实时保证系统数据的强一致性。

# 二　分布式事务

简单的说，就是一次大的操作由不同的小操作组成，这些小的操作分布在不同的服务器上，且属于不同的应用，分布式事务需要保证这些小操作要么全部成功，要么全部失败。本质上来说，分布式事务就是为了保证不同数据库的数据一致性。

